A multiobjective evolutionary algorithm to find community structures based on affinity propagation

Shang, Ronghua; Luo, Shuang; Zhang, Weitong; Stolkin, Rustam; Jiao, Licheng

DOI:
10.1016/j.physa.2016.02.020

Accepted Manuscript

A multiobjective evolutionary algorithm to find community structures based on affinity propagation

Ronghua Shang, Shuang Luo, Weitong Zhang, Rustam Stolkin, Licheng Jiao

PII: S0378-4371(16)00186-2
DOI: http://dx.doi.org/10.1016/j.physa.2016.02.020
Reference: PHYSA 16914

To appear in: *Physica A*

Received date: 13 July 2015
Revised date: 18 December 2015

Please cite this article as: R. Shang, S. Luo, W. Zhang, R. Stolkin, L. Jiao, A multiobjective evolutionary algorithm to find community structures based on affinity propagation, *Physica A* (2016), http://dx.doi.org/10.1016/j.physa.2016.02.020

A Multiobjective Evolutionary Algorithm to Find Community Structures Based on Affinity Propagation

Ronghua Shang[a], Shuang Luo[a], Weitong Zhang[a], Rustam Stolkin[b] and Licheng Jiao[a]

([a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, China; [b] School of Mechanical Engineering, University of Birmingham, UK)

Abstract: Community detection plays an important role in reflecting and understanding the topological structure of complex networks, and can be used to help mine the potential information in networks. This paper presents a Multiobjective Evolutionary Algorithm based on Affinity Propagation (APMOEA) which improves the accuracy of community detection. Firstly, APMOEA takes the method of affinity propagation (AP) to initially divide the network. To accelerate its convergence, the multiobjective evolutionary algorithm selects nondominated solutions from the preliminary partitioning results as its initial population. Secondly, the multiobjective evolutionary algorithm finds solutions approximating the true Pareto optimal front through constantly selecting nondominated solutions from the population after crossover and mutation in iterations, which overcomes the tendency of data clustering methods to fall into local optima. Finally, APMOEA uses an elitist strategy, called "external archive", to prevent degeneration during the process of searching using the multiobjective evolutionary algorithm. According to this strategy, the preliminary partitioning results obtained by AP will be archived and participate in the final selection of Pareto-optimal solutions. Experiments on benchmark test data, including both computer-generated networks and eight real-world networks, show that the proposed algorithm achieves more accurate results and has faster convergence speed compared with seven other state-of-art algorithms.

Keywords: Complex network; community detection; affinity propagation; multiobjective evolutionary algorithm

1. Introduction

Complex networks are used to represent the relationship between different individuals in a system, such as biological networks, social networks, information transmission networks and so on. In a network, a node represents an individual in a system, and the line connecting two nodes represents the relationship between these two individuals. Complex networks have some basic statistical properties, such as "small world effect [1]" and "scale-free [2]". Another important property is community structure [3-5]. Nodes in the same community are more closely connected with each other than with other nodes in different communities. Community detection is significant in revealing the inherent community structure and analyzing the function of complex networks.

In the past decades, numerous algorithms have been proposed since the significance of mining the community structure of complex networks was realized. Example methods include GN [4], proposed by Girvan and Newman, which is one of the most classical hierarchical clustering algorithms [4-6]. Taking Kernighan-Lin algorithm [7] and

spectral bisection algorithm [8] as their representatives, graph partitioning methods try to divide the whole network into a few sub-graphs. Besides, some methods are based on the node similarity [9-10], while others are proposed to find overlapping communities in networks [11-13].

However, these algorithms still have much room for improvement in increasing the accuracy of detection results. Thus, this paper proposes an algorithm which combines a powerful data clustering method, Affinity Propagation (AP) [31], and an evolutionary algorithm to achieve higher detection results within a few iterations. Firstly, this paper employs efficient data clustering methods, such as AP, to preprocess the networks. First a data clustering method is used to obtain initial partition results within a few steps. Next, these results are screened and an evolutionary algorithm, which has the characteristic of global optimization, is used to further improve these results. We demonstrate that this combination of methods produces good results on both real and computer generated network benchmark data.

The remaining parts of this paper are arranged as follows. Section 2 describes related work. Section 3 explains the background and design of the APMOEA algorithm. Section 4 presents experimental results and analysis. Section 5 provides concluding remarks.

2. Related works

In recent years, methods based on the optimization of modularity Q have also been widely used [14-19], such as the Newman greedy algorithm [16], simulated annealing [14] and external optimization [18]. However, it has been proved that maximizing modularity Q is computationally intractable [20]. Because evolutionary algorithms have several good characteristics, there is increasing interest in combining them with other methods in solving this discrete optimization problem. Firstly, objective functions can be optimized by the evolutionary algorithm regardless of whether or not they are continuous. Secondly, the method of multipoint searching in evolutionary algorithm ensures a good global searching ability. Thirdly, evolutionary algorithms have strong combination ability with other algorithms. In 2008, Clara Pizzuti proposed an algorithm named GA-Net [21], which introduced a new objective function termed Community Score (*CS*) and employed an evolutionary algorithm as the optimization method. However, the *CS* criteria only measures the degree of intra-connections in communities without regarding that of inter-connections between communities. Consequently, in 2009 Clara Pizzuti proposed MOGA-Net [22], which is a multiobjective evolutionary algorithm employing the method of NSGA-II [23] and introduces the concept of Community Fitness (*CF*), which is complementary to *CS*, as the second objective function. Through maximizing these two objectives simultaneously, the algorithm obtained a set of solutions which revealed community structure at different hierarchical levels. In 2011, Gong et al. proposed a memetic algorithm (Meme-Net) to detect community structure in networks [24]. The algorithm employed modularity density D [25] as the optimization function and used a hill-climbing strategy as its search strategy. By adjusting the parameter in the objective function D, Meme-Net could find better partitions in networks at different resolutions. On that basis,

Shang et al. proposed a method based on Modularity and Improved Genetic Algorithm (MIGA) in 2013 [26]. MIGA adopted modularity *Q* [3] instead of modularity density *D* as the objective function to simplify the algorithm and reduced its computational complexity. Meanwhile, using prior information in population initialization makes the algorithm more targeted and accurate in community detection. Moreover, MIGA took simulated annealing as the search algorithm, which has greatly improved the ability of local search. In 2012, Gong et al. put forward an algorithm named MOEA/D-Net [27] based on MOEA/D [29]. It decomposed a two-objective optimization problem into several scalar optimization sub-problems and optimized them simultaneously to obtain a set of solutions approximating the true Pareto-optimal front. Another algorithm, CCDECD [30], which first integrates Cooperative Co-evolution framework into Differential Evolution based Community Detection has been proposed to decompose a network into several smaller parts and optimize them respectively. Abias grouping scheme is employed as the pre-processing method to improve the accuracy of the partition results. Through combining these effective strategies, this method can achieve promising results on larger networks.

However, evolutionary algorithms have some drawbacks, such as low convergence speed, premature convergence and degradation. Aiming at solving some of these problems, this paper presents a Multiobjective Evolutionary Algorithm based on Affinity Propagation (APMOEA). Firstly, the proposed algorithm uses Affinity Propagation (AP) method for the preliminary partition of networks. AP algorithm [31] is a data clustering method proposed by Frey and Dueck in 2007. Through passing messages between data points until a high-quality set of exemplars and corresponding clusters finally emerges, the method of AP possesses high stability and accuracy without the need to know the exact number of clusters in advance. Nevertheless, its clustering results depend heavily on the selection of a parameter which is referred to as “Preference” (*P*) [31]. Setting the value of parameter *P* too high or too low will lead to poor results. Therefore, to guarantee the effectiveness of the clustering results, a certain number of parameter *P* will be initialized randomly within a range, which is selected according to an experiment (details shown in section 4.3), and the corresponding clusters will be obtained by AP algorithm. Some nondominated solutions will be chosen from the above results as the preliminary partitioning results of networks. However, AP algorithm may not always achieve the optimal clustering results as it is limited to the shape of the data structure [32-33]. Therefore, in order to improve the accuracy of the final results, a multiobjective evolutionary algorithm is employed here for a further search. The preliminary partitioning results will be taken as the initial population, and crossover and mutation operators are iterated to update the set of nondominated solutions. Due to the high quality of the initial population obtained by the AP algorithm, it only needs a few iterations for the algorithm to quickly converge to the optimal solutions. In addition, an elitist strategy, known as “external archive”, is adopted to settle the problem of degradation caused by random search in evolutionary algorithm. This means that the preliminary partitioning results will be stored as elitist solutions and participate in the selection of Pareto-optimal solutions with the set of nondominated solutions obtained by the evolutionary algorithm, improving stability. Simulation results on artificial and real-world networks show that the proposed algorithm converges faster and is more accurate compared with other state-of-art algorithms.

3. The proposed algorithm

The main parts of APMOEA consist of the choice of objective functions, selection method for nondominated solutions, the way that uses AP to get the preliminary partitions of networks and the genetic operators in multiobjective evolutionary algorithm. In the following sections the above contents will be introduced in details. The procedure of APMOEA is shown in TABLE 1.

TABLE1: The procedure of APMOEA

Algorithm 1: APMOEA
Input: Affinity matrix of network: $\boldsymbol{A}$; Population size of parameter P: N_{umP}; Crossover possibility: p_c; Mutation possibility: p_m; Maximum number of iterations: G_{max};
Output: A set of Pareto-optimal solutions;
Step1: Get the preliminary partitions $\boldsymbol{C}_{pre}$ by using AP method; Archive $\boldsymbol{C}_{pre}$; *loop*:=1
Step2: Chromosomes $\boldsymbol{C}_{child} \leftarrow$ Genetic operation($\boldsymbol{C}_{pre}$, p_c, p_m);
Step3: $f_1(\boldsymbol{C}_{child})$, $f_2(\boldsymbol{C}_{child}) \leftarrow$ Objective function f_1, f_2 of $\boldsymbol{C}_{child}$; Update Pareto-optimal front $\boldsymbol{C}_{optimal}$ through selecting nondominated solutions from $\boldsymbol{C}_{child}$;
Step4: If *loop*=G_{max}, go to Step5; Otherwise, *loop*:=*loop*+1, return to Step2.
Step5: Selecting nondominated solutions from $\boldsymbol{C}_{optimal}$ and $\boldsymbol{C}_{pre}$ as final Pareto-optimal solutions and output a set of Pareto-optimal solutions.

3.1. Background of the proposed algorithm

3.1.1. Affinity Propagation method

In 2007 Frey and Dueck proposed a powerful clustering method termed as Affinity Propagation [31], which has shown its high efficiency in various fields. It has not only low error rate as well as strong stability, but also short running time. Furthermore, AP does not need to specify the number of clusters in advance before clustering. The basic idea of AP is relatively simple. Initially, it takes negative real-valued similarities between pairs of data points as input, where $s(i, k)$ indicates how appropriate it is for data point k to be the exemplar for data point i. The algorithm considers all data points as potential exemplars at the beginning and transmits messages between data points until a set of high-quality exemplars and corresponding clusters gradually emerge. There are two types of messages. One called "responsibility" $r(i, k)$, representing the possibility that point k is selected as the exemplar for point i. The other is "availability" $a(i, k)$, representing how appropriate it is for point i to choose point k as its exemplar. The overall process of message transmission can be expressed by the following formulae:

$$r^{(t+1)}(i,k) \leftarrow (1-\lambda)\left(s(i,k) - \max_{k's.t.k'\neq k}\{a(i,k') + s(i,k')\}\right) + \lambda r^{(t)}(i,k) \tag{1}$$

$$a^{(t+1)}(i,k) \leftarrow (1-\lambda)\left(\min\left\{0, r(k,k) + \sum_{i's.t.i'\notin\{i,k\}} \max\{0, r(i',k)\}\right\}\right) + \lambda a^{(t)}(i,k) \tag{2}$$

$$a^{(t+1)}(k,k) \leftarrow (1-\lambda)\left(\sum_{i's.t.i'\neq k} \max\{0, r(i',k)\}\right) + \lambda a^{(t)}(k,k) \tag{3}$$

where parameter λ is a damping factor [31] for the prevention of numerical oscillations and its value is between 0 and 1. Before iterations, the values of "responsibility" and "availability" should be set to zero, which can be represented as $r^{(0)}(i, k)=0$, $a^{(0)}(i, k)=0$. AP takes as input the value of $s(k, k)$ for every data point to weight how likely they are to be chosen as exemplars. These parameters are known as "preferences" (P). As all data points can be regarded as potential exemplars during initialisation, the preferences share a common value, which is usually the median or minimum of negative similarity matrix $\boldsymbol{S}$.

3.1.2. Multiobjective optimization

A multiobjective optimization problem with q objectives can be defined as [29] [34]:

$$\max \boldsymbol{F}(\boldsymbol{x}) = \left\{ f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_q(\boldsymbol{x}) \right\} \tag{4}$$

where $\boldsymbol{x}=(x_1, x_2, ..., x_n) \in Z$ is the decision vector, and Z is the feasible region in decision space. Given two decision vectors $\boldsymbol{x}, \boldsymbol{x}^* \in Z$, $\boldsymbol{x}^*$ is said to dominate $\boldsymbol{x}$ (denoted as $\boldsymbol{x}^* \succ \boldsymbol{x}$) if and only if:

$$\left(\forall i \in \{1,2,\cdots,q\}: f_i(\boldsymbol{x}^*) \geq f_i(\boldsymbol{x})\right) \wedge \left(\forall j \in \{1,2,\cdots,q\}: f_j(\boldsymbol{x}^*) > f_j(\boldsymbol{x})\right) \tag{5}$$

If in feasible region Z, there exists no decision vector $\boldsymbol{x}$ such that $\boldsymbol{x} \succ \boldsymbol{x}^*$, we call $\boldsymbol{x}^*$ a Pareto-optimal solution or nondominated solution. All these Pareto-optimal solutions compose the Pareto-optimal set and its corresponding figure plotted in the objective space is called the Pareto-optimal front. Thus, the goal for multiobjective optimization is to find a set of solutions approximating the true Pareto-optimal front.

Different from the single objective optimization, multiobjective optimization can achieve a group of nondominated solutions in a single run, and reveals the hierarchical structure of networks to meet different needs for division. Note that the optimal solutions found by single objective optimization are usually included in the Pareto-optimal set [35]. In the following sections, we will give experiments to illustrate the advantages of the multiobjective optimization algorithms over the single objective optimization algorithms.

3.2. Objective functions

Objective functions that are commonly used in community detection can be summarized as follows: modularity Q, modularity density D, community score CS and community fitness CF. Modularity Q [3] is a widely used standard put forward by Girvan and Newman, and the solution with higher value of Q indicates the better partitioning of a network. The definition of modularity Q can be formulated as follows:

$$Q = \sum_{s=1}^{K} \left[\frac{l_s}{m} - \left(\frac{d_s}{2m} \right)^2 \right] \tag{6}$$

Where l_s represents the number of edges connecting all nodes in community s, m is the total number of edges in network. d_s is the sum of degrees of all the nodes in community s. The higher the value of Q is, the denser the connection within a community.

Although many optimization algorithms based on modularity Q have recently emerged [36], they suffer from

a problem of resolution limit such that small clusters can often fail to be separated from larger clusters. To avoid this problem, the proposed algorithm adopts modularity density D [25], which has yielded significant improvement over modularity Q, as an objective function.

Consider an undirected network $G=(V, E)$ with vertex set V and edge set E. Its adjacency matrix is $\boldsymbol{A}$. If there exists a connection between node i and node j, $A_{ij}=1$; otherwise $A_{ij}=0$. If V_1 and V_2 are two disjoint subsets of V, then there will be $L(V_1,V_2)=\sum_{i\in V_1, j\in V_2} A_{ij}$ and $L(V_1,\overline{V_1})=\sum_{i\in V_1, j\in \overline{V_1}} A_{ij}$, where $\overline{V_1}=V-V_1$. For a given partition $\Omega=\{V_1, V_2, ..., V_m\}$, V_i is the vertex set of subgraph G_i. For $i=1, 2, ..., m$, the modularity density D can be expressed as:

$$D=\sum_{i=1}^{m}\frac{L(V_i,V_i)-L(V_i,\overline{V_i})}{|V_i|} \tag{7}$$

In the proposed algorithm, the equation is divided into two parts as two objectives for optimization. The first part, labeled as ratio association [37], indicates how closely nodes connect with each other in the same community. The second part, known as ratio cut [38], indicates how closely nodes connect with others in different communities. Maximizing modularity density D can find communities with dense intra-connections and sparse inter-connections, which suggests an optimal partition of a network. Thus, the two-objective optimization problem can be formulated as a maximum optimization problem:

$$\begin{cases}\max\ f_1(x)=\sum_{i=1}^{m}\frac{L(V_i,V_i)}{|V_i|}\\ \max\ f_2(x)=-\sum_{i=1}^{m}\frac{L(V_i,\overline{V_i})}{|V_i|}\end{cases} \tag{8}$$

3. 3. The selection method for nondominated solutions

In this paper, the method proposed in NSGA-II [23] is employed here to select the nondominated solutions. It consists of two aspects: fast nondominated sorting approach and crowded-comparison approach.

Firstly, we employ the fast nondominated sorting approach to sort population S_g into different nondomination levels and choose only individuals of the first nondominated front. The updated population is recorded as $S_{g\text{-}Pareto}$. Then, to obtain a better spread of Pareto-optimal front, the solutions will be screened again by the crowded-comparison approach [23]. For a given individual $g\in S_{g\text{-}Pareto}$, its crowding-distance can be measured by the following formula [34]:

$$d\left(g,S_{g-Pareto}\right)=\sum_{k=1}^{q}\frac{d_k\left(g,S_{g-Pareto}\right)}{f_k^{\max}-f_k^{\min}} \tag{9}$$

where, $f_k^{\max}$ and $f_k^{\min}$ represent the maximum and minimum value of the kth objective respectively, and q stands for the number of objective functions. $d_k(g, S_{g\text{-}Pareto})$ which can be expressed as:

$$d_k\left(g,S_{g-Pareto}\right)=\begin{cases}\infty & ,if\ f_k(g)=M\ or\ m\\ \min\left\{f_k\left(g_j\right)-f_k(g_i)\right\}, & \text{others}\end{cases} \tag{10}$$

where, M and m are the maximum and minimum value of kth objective found in $S_{g\text{-}Pareto}$, g_i and g_j are subjected to:

$\{f_k(g_i) < f_k(g) < f_k(g_j) \mid g_i, g_j \in S_{g\text{-}Pareto}\}$. From formula (9) we can see that solutions with greater crowding-distance make more contributions in improving the diversity of the population. Hence, according to their corresponding values of crowding-distance, the solutions will be updated by removing some individuals that are too crowded in the Pareto-optimal front.

3. 4. The preliminary partition by AP method

Data clustering methods such as *K*-means [39] have fast convergence speeds, but are very sensitive to the choice of initial clustering centers and require prior knowledge about the number of clusters, which is typically unavailable in real world community detection problems. Compared to *K*-means, the AP clustering method is more precise and stable. More importantly, there is no need for the AP method to know the number of clusters in advance. Since it was first proposed in 2007, some scholars have applied the AP algorithm to community detection [40-43].

Community detection is supposed to be a graph clustering problem [20] [44], in which a network can be viewed as a big graph that is made up of several subgraphs, and connections are much denser within the same subgraph than between different subgraphs. In data clustering, the comparison between two samples actually means the comparison between the same attributes that belong to them. However, we can only know the topological information of networks in community detection as a graph clustering problem. It is critical to choose a high-quality similarity measure to transform community detection into a data clustering problem. In light of comparative experimental results in the literature [45], the proposed algorithm employs a similarity measure based on the signaling process [46], which has proven its high accuracy.

The Similarity measure based on the signaling process was proposed by Hu et al. in 2008 [46]. The essential principle of this method regards a network with n nodes as a signal transmission system, in which every node can send, receive and record signals. After a period of transmission, the distribution of signals over the whole network produced by the vertices in the same community will be similar. The signaling process can be expressed as:

$$\boldsymbol{W} = (\boldsymbol{I}_n + \boldsymbol{A})^t \tag{11}$$

where, $\boldsymbol{I}_n$ is an n-dimensional identity matrix and $\boldsymbol{A}$ represents the adjacency matrix of the network, t is the transmission time, which takes a value of 3 in this paper.

Supposing an undirected network with n nodes, its adjacency matrix is $\boldsymbol{A}$, at first we can compute the signal transmission matrix $\boldsymbol{W}=(\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_k, ..., \boldsymbol{w}_n)^{\mathrm{T}}$, where $\boldsymbol{w}_k=(w_{k1}, w_{k2}, ..., w_{kn})$, k=1, 2, ..., n. Here $\boldsymbol{w}_k$ indicates the effect on n nodes produced by the kth node after t steps. In order to get comparable results, we should normalize every row vector in matrix $\boldsymbol{W}$. Different from the original normalization method mentioned in [46], the matrix after normalization is recorded as $\boldsymbol{U}=(\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_k, ..., \boldsymbol{u}_n)^{\mathrm{T}}$, where $\boldsymbol{u}_k=(u_{k1}, u_{k2}, ..., u_{kn})$, k=1, 2, ..., n, and $\boldsymbol{u}_{kl}$ is subject to:

$$u_{kl} = w_{kl} / \sqrt{\sum_{j=1}^{n} w_{kj}^2} \tag{12}$$

where, l=1, 2, ..., n. Following these procedures, we can transform the topology information of the network into geometrical information of vectors in an n-dimensional Euclidian space. It is worth noting that in order to apply the

AP algorithm for clustering, we have to compute negative Euclidean distance between pairs of n vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n$ for obtaining the negative similarity matrix $\boldsymbol{S}$.

According to the descriptions above, the detailed procedure of using AP method for the preliminary partitioning of networks is shown in TABLE 2:

TABLE2: The preliminary partitioning of networks by AP method

Algorithm 2: The preliminary partitioning of networks by AP method
Input: Affinity matrix of network: $\boldsymbol{A}$; Population size of parameter P: N_{umP}; Maximum size of dominant population: N_{max};
Output: The preliminary partitioning results $\boldsymbol{C}_{pre}$;
Step1: The negative similarity matrix $\boldsymbol{S} \leftarrow$ Signal similarity($\boldsymbol{A}$);
Step2: Population $\boldsymbol{C}_P \leftarrow$ Initialize parameter $P(N_{umP})$;
Step3: Population $\boldsymbol{C}_{AP} \leftarrow$ Affinity propagation($\boldsymbol{S}$, $\boldsymbol{C}_P$);
Step4: $f_1(\boldsymbol{C}_{AP})$, $f_2(\boldsymbol{C}_{AP}) \leftarrow$ Objective function f_1, f_2 of $\boldsymbol{C}_{AP}$;
Step5: $\boldsymbol{C}_{pre} \leftarrow$ Selection ($\boldsymbol{C}_{AP}$, $f_1(\boldsymbol{C}_{AP})$, $f_2(\boldsymbol{C}_{AP})$, N_{max}); Output $\boldsymbol{C}_{pre}$.

3. 5. Further search using multi-objective evolutionary algorithm

In order get solutions approximating the true Pareto-optimal front and converge to the global optimum, here the proposed algorithm takes multi-objective evolutionary algorithm (MOEA) as a measure for a further search. Through crossover and mutation on the preliminary partitioning results obtained by AP method, the diversity of the solution space will be greatly increased and it is helpful for avoiding local optima.

According to the number of the objective functions, evolutionary algorithms can be divided into two categories: single objective evolutionary algorithms and multi-objective evolutionary algorithms. However, compared to the multi-objective evolutionary algorithm, the single objective evolutionary algorithm gets only one definite solution rather than a group of solutions in one run, which is not conducive for finding the true partitions. Thus, this paper adopts the multi-objective evolutionary algorithm as a further search method.

3.5.1. Representation and initialization

For each partition of a network with n nodes, we use a string with n integer numbers as its representation. Such as a partition $\boldsymbol{x}$:

$$\boldsymbol{x}=[x^1\ x^2 ...\ x^i ...\ x^n] \tag{13}$$

Here x^i is a class label that represents the cluster node to which i belongs. Nodes in the same cluster have the same class label. For example, if node 1 and node 2 are in the same cluster, then $x^1=x^2$.

Generally, in community detection problems, population initialization is typically performed by generating a group of partitions randomly. Although this approach is simple and fast, it takes many iterations for the algorithm to converge to the optimal results. In the proposed algorithm, we employ a set of good partitioning results obtained by the AP method as the initialization population of the evolutionary algorithm, which has greatly enhanced the quality of population initialization and thus promotes rapid convergence to the optimal solution.

3.5.2. Genetic operators

For the sake of increasing the diversity of the solution space and finding solutions approximating the true Pareto-optimal front, we use the crossover and mutation operations in the process of evolution. They are introduced respectively as follows.

Crossover: Conventional methods such as one-point crossover or two-point crossover are simple to operate but, considering the phenotypic characteristics of the chromosomes, they are not suitable for the proposed algorithm as they may destroy some useful genetic information inherited from the parents. To generate offspring carrying features common to their parents, here we employ a two-way crossover operation [23].

For example, for a network of 5 nodes, two chromosomes $\boldsymbol{r}_a$=[1 2 1 1 3] and $\boldsymbol{r}_b$=[2 3 3 4 2] are selected randomly from the parent population and their corresponding offspring generated by crossover operation are $\boldsymbol{r}_c$ and $\boldsymbol{r}_d$. If we select the one-point crossing and choose the third node as the crossover point, then all the genes in chromosomes $\boldsymbol{r}_a$ and $\boldsymbol{r}_b$ will be exchanged after that point. As the specific process shown in TABLE 3, the offspring are $\boldsymbol{r}_c$=[2 3 1 1 3] and $\boldsymbol{r}_d$=[1 2 3 4 2]. It's clear to see that node 1, 3 and 4 should be in the same community originally in chromosome $\boldsymbol{r}_a$, however, they are assigned to totally different communities in chromosome $\boldsymbol{r}_d$, which has destroyed the original information of the parent. If we choose two-way crossover this time and the third node is still the crossover point, the genes whose value are the same as the value of the third node will be retained, namely the first, the third and the fourth genes in $\boldsymbol{r}_a$, and the second and the third genes in $\boldsymbol{r}_b$. The rest of the genes will be swapped. This process is shown in TABLE 4. The results $\boldsymbol{r}_c$=[1 3 1 1 2] and $\boldsymbol{r}_d$=[1 3 3 1 3] successfully inherit effective information from their parents.

TABLE 3: One-point crossing

v		$\boldsymbol{r}_a$		$\boldsymbol{r}_b$		$\boldsymbol{r}_c$	$\boldsymbol{r}_d$		$\boldsymbol{r}_a$		$\boldsymbol{r}_b$		**v**
1		1		2		2	1		1		2		1
2		2		3		3	2		2		3		2
③	→	①	→	3	→	①	③	←	1	←	③	←	③
4		①	→	4	→	①	④	←	1	←	④		4
5		③	→	2	→	③	②	←	3	←	②		5

TABLE4: Two-way crossing

v		$\boldsymbol{r}_a$		$\boldsymbol{r}_b$		$\boldsymbol{r}_c$	$\boldsymbol{r}_d$		$\boldsymbol{r}_a$		$\boldsymbol{r}_b$		**v**
1		①	→	2	→	①	1		1		2		1
2		2		3		3	③	←	2	←	③		2
③	→	①	→	3	→	①	③	←	1	←	③	←	③
4		①	→	4	→	①	1		1		4		4
5		3		2		2	3		3		2		5

Mutation: in this paper, we adopt the following mutation mode: randomly select a gene of a chromosome, and change its value to an integer in the set of {1, 2, ..., L}, where L is the largest class number in that chromosome. It is easy to operate and helps increase the diversity of the population. Besides, invalid mutation will be effectively avoided through limiting the scope of mutation. For each of the chromosomes to be mutated, 20% of the genes will be selected for mutation. For example, if the chromosome $\boldsymbol{r}_m$=[1 2 1 1 3] is selected, then L should equal to 3.

Select 20% genes (namely one gene) randomly, assuming it is the fourth vertex, then its corresponding value of gene can be turned into any one among 1, 2 and 3. This procedure is shown in TABLE 5.

TABLE5: Mutation operation

v	r_m	r_m'
1	1	1
2	2	2
3	1	1
④ →	① →	③
5	3	3

3. 6. Elitist strategy of external archive

An elitist strategy known as external archive is used here as an offset with regard to the problem of degradation that emerges in the evolutionary algorithm. External archive is similar to the elitist strategy proposed in [47]. As the preliminary partitions obtained by the AP method are a group of superior solutions, they will be archived additionally as the elitists. After a new set of nondominanted solutions being found by a further search using the evolutionary algorithm, they will be incorporated with the archived solutions as a whole, from which the final Pareto-optimal set is selected. This can ensure the dominance of solutions and prevent the degradation of final results to a certain extent.

3. 7.The flow chart of APMOEA

In light of the above analysis, the flow chart of APMOEA is shown in Fig.1:

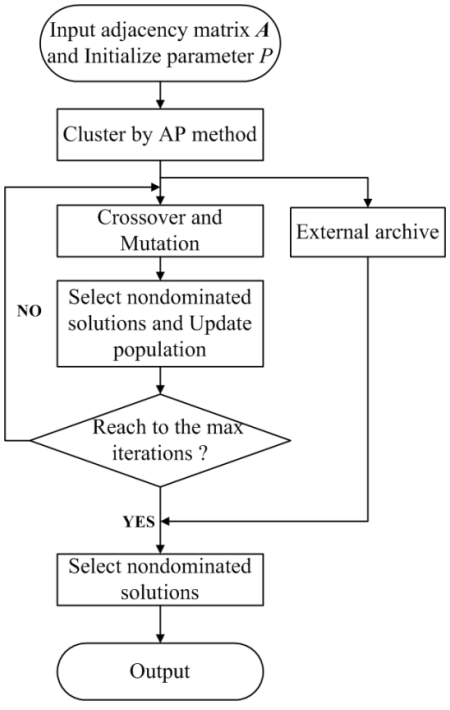


Fig.1. The flow chart of APMOEA

4. Experimental results and analysis

In order to verify the effectiveness of the proposed algorithm, this paper selects several evolutionary algorithms, such as GA algorithm, Meme-Net [24], MIGA [26], MOEA/D-Net [27], MODPSO algorithm [28] and some classical algorithms like Infomap[54] and FastNewman[16] for comparison. In addition, we also make comparisons about the proposed algorithm and a single-objective evolutionary algorithm based on the preliminary partitions attained by AP algorithm. In the following sections, we will introduce the related networks for simulation, evaluation metrics, the analysis of experimental results and the average total time costs of the proposed algorithms.

4. 1. Networks for simulation

4.1.1. Computer-generated networks

This network is proposed by Lancichinetti et al. [48] based on the classic benchmark network [4]. They consider that distributions of community size and degree are power laws, with exponents τ_1 and τ_2, respectively. Each node shares a fraction 1-μ of its links with nodes in the same community and a fraction μ with nodes in other communities. With the increase of parameter μ, the structure of the communities in networks becomes fuzzier, and it is more difficult to find the true partitions.

4.1.2. Real-world networks

In the experiment, eight real-world networks are used to test the proposed algorithm. These networks are described as follows.

The Zachary's karate club [49] put forward by Zachary has 34 nodes, representing 34 members of the club. This club split into 2 parts by chance. It has a total number of 78 edges connecting the nodes.The Bottlenose Dolphins network [50] was proposed by Lusseau on the basis of 7 years observation on 62 dolphins in New Zealand. The dolphins are divided into 2 categories. There are 159 edges in the network. The American college football network [4] consists of 115 nodes and 616 edges. Each node represents an American college football team, and each edge represents a match between the two football teams being connected. The network is divided into 12 categories. The Books about US politics network represents 105 American political books on sale at Amazon.com. This network associate the books brought by the same buyers. It was divided into 3 categories by Newman [15]. In contrast to the above 4, the true partitions of the following are unknown. These networks are SFI [4], netscience [53], Power grid [1] and PGP [52]. SFI consists of 118 vertices, which represent the largest component of the Santa Fe Institute collaboration network. There are about 200 edges in this network. The netscience network is a coauthorship network of 1589 scientists working on network theory and experiments. As this paper deals with only unweighted networks, we will regard this weighted network as unweighted one. The Power grid network is an undirected, unweighted network representing the topology of the Western States Power Grid of the United States, in which 4941 power base stations transmit electricity through 6594 transmission lines. The second largest network tested in this paper is PGP, the giant component of the network of users of the Pretty-Good-Privacy algorithm for secure information interchange. This network consists of 10680 nodes and 24340 edges. The final network used in

our experiments is the Internet network which contains 22963 nodes and 48436 edges which is a snapshot of the structure of the Internet network.

4. 2. Evaluation metrics

In order to evaluate the similarity between the solutions and the true partitions, here we take the modularity density D, introduced in section 3.2, as the evaluation index.

We also use Normalized Mutual Information (NMI) [51] as an evaluation index. Supposing h_1 is the true partition of a network and h_2 is a partition found by us. $\boldsymbol{H}$ is the confusion matrix, where H_{ij} represents the number of the same nodes that exist both in community i of partition h_1 and in community j of partition h_2. The definition of NMI is as follows:

$$I(h_1,h_2)=\frac{-2\sum_{i=1}^{H_{h_1}}\sum_{j=1}^{H_{h_2}}H_{ij}\log\left(H_{ij}N/H_{i.}H_{.j}\right)}{\sum_{i=1}^{H_{h_1}}H_{i.}\log\left(H_{i.}/N\right)+\sum_{j=1}^{H_{h_2}}H_{.j}\log\left(H_{.j}/N\right)} \tag{14}$$

Here H_{h1} (H_{h2}) is the total number of the communities in h_1 (h_2), $H_{i.}$ is the sum of elements of $\boldsymbol{H}$ in row, while $H_{.j}$ is sum of elements of $\boldsymbol{H}$ in column. When h_1=h_2, then $I(h_1, h_2)$=1; otherwise, if there is no common vertex in both partitions, $I(h_1, h_2)$=0.

Another evaluation index we employ here, to estimate the partition results of the networks whose true partitions are unknown, is modularity Q, which has been introduced in Section 3.1. The higher the value of Q, the better result it usually indicates. Usually, the value of Q ranges from 0.3 to 0.7 in practice.

4.3. Parameter selection in APMOEA

This section describes experiments conducted to choose appropriate values for parameters that are employed in the proposed algorithm. Appropriate parameter choices are important for achieving good results. These parameter values are tested on the extension of GN Benchmark networks [48] and six real-world networks including the karate club network, the dolphins network, the football network, the books about US politics network, the SFI network and the netscience network.

As introduced in section 2.1, the clustering results obtained by AP method depend mainly on the selection of parameter P. Generally speaking, the higher the value of parameter P, the more clusters will be found, and vice versa [31]. Different networks have different number of communities. Therefore, how to set the value of P is an important problem.

In the practice, the value of P is usually set to be the median of the negative similarity matrix $\boldsymbol{S}$, which can vary since the similarity matrix $\boldsymbol{S}$ is different in various networks. To choose an appropriate value range of parameter P, we perform an experiment on the extension of GN Benchmark networks in which parameter μ changes from 0.00 to 0.50 with an interval of 0.1 and P is set to range from -20 to 20 with an interval of 1. The corresponding clustering results NMI are shown in Fig. 2.

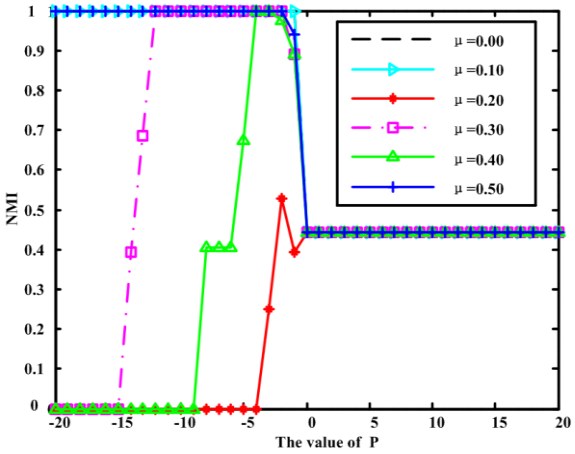


Fig.2 The results NMI on the extension of GN Benchmark networks with different values of parameter *P*

Similarly, the corresponding clustering results of NMI on four real-world networks for which the true partitions are known are shown in Fig. 3. And the results of Q and D on two real-world networks where the true partition is unknown are shown in Fig. 4.

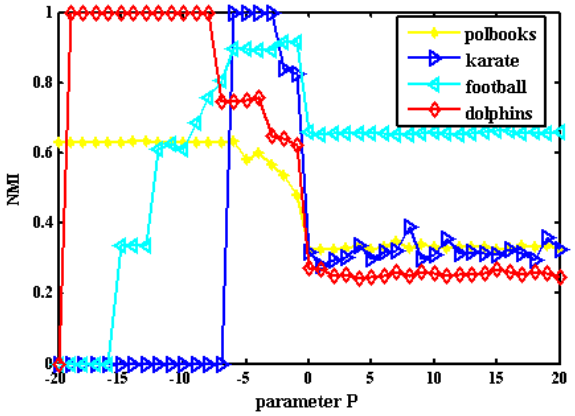


Fig.3 The results of NMI of four real-world networks with different values of parameter *P*

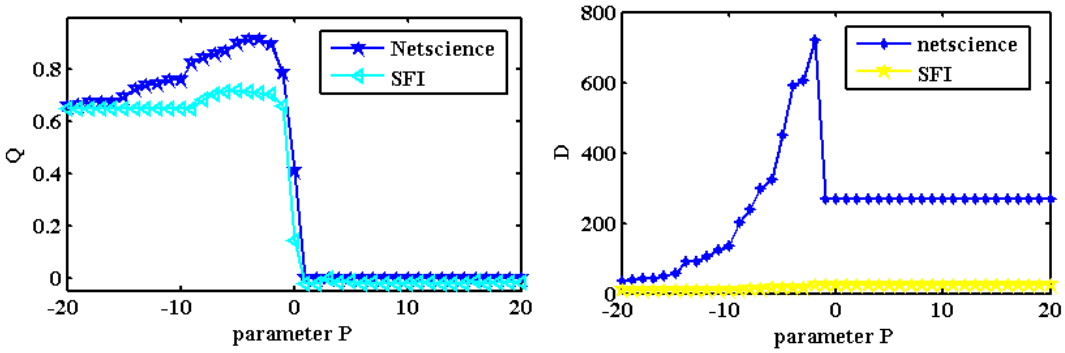


Fig.4 The results of Q and D of two real-world networks with different values of parameter *P*

It can be seen from Fig.2, Fig.3 and Fig.4 that, with the increasing of *P* from -20 to 20, the accuracy of clustering results gradually rises to a maximum, and then declines to the minimum when the value of *P* is beyond a certain boundary. Only when *P* is in the range of [-10, 0] can AP algorithm achieve good results. Thus the initialization range of *P* will be selected as [-10, 0] in the proposed algorithm.

Apart from the parameter *P*, the initial population size *popsize*, the max iteration number G_{max} and the

maximum number of dominant solutions *NM* also have an important influence on the performance of the algorithm. In each experiment, we will select their appropriate values by changing the value of one parameter and fixing the others. Fig.5 shows the max NMI values over 30 runs of the extension of GN Benchmark networks in which parameter μ changes from 0.00 to 0.50 with an interval of 0.05 varies with different values of *popsize* when choosing the max iteration number G_{max}=30 and the maximum dominant solution number *NM*=40. Fig.6 and Fig.7 shows the results NMI and Q and D of six real-world networks respectively.

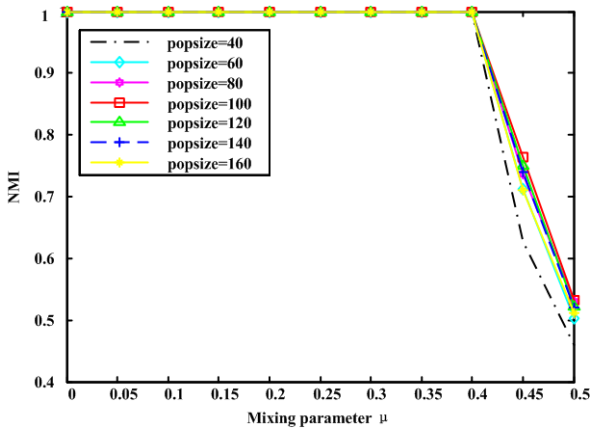


Fig.5 The results of NMI of the extension of GN Benchmark networks with different population value

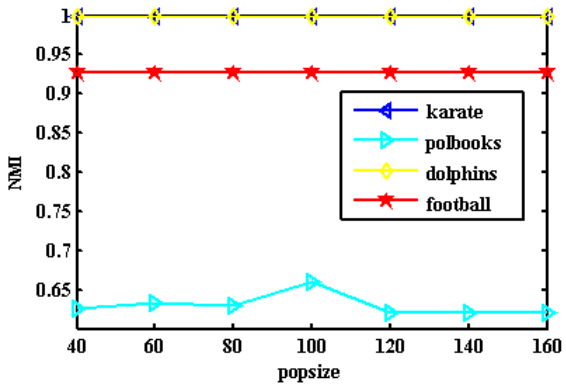


Fig.6 The results of NMI of four real-world networks with different population value

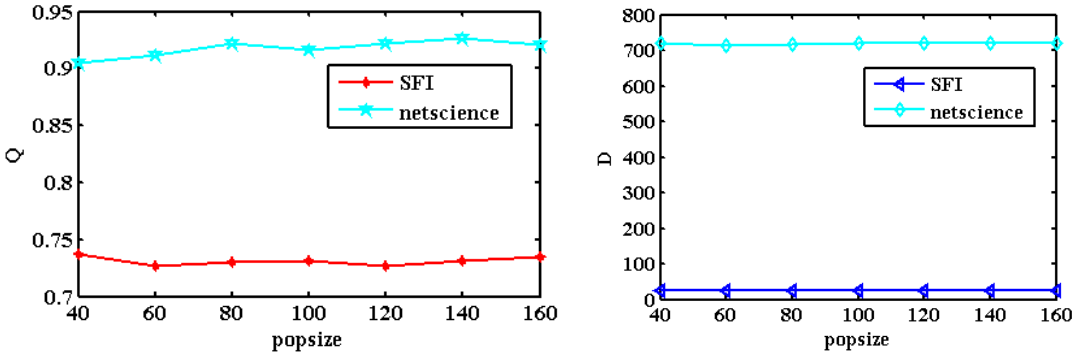


Fig.7 The results of Q and D of two real-world networks with different population value

It can be seen from Fig.5, Fig.6 and Fig.7 that, the algorithm does not achieve good results if the value of *popsize i*s too small. When the population reaches more than 80, the difference of the results are not significant and the best NMI, Q and D values can be obtained when *popsize*=100, which is selected in the proposed algorithm.

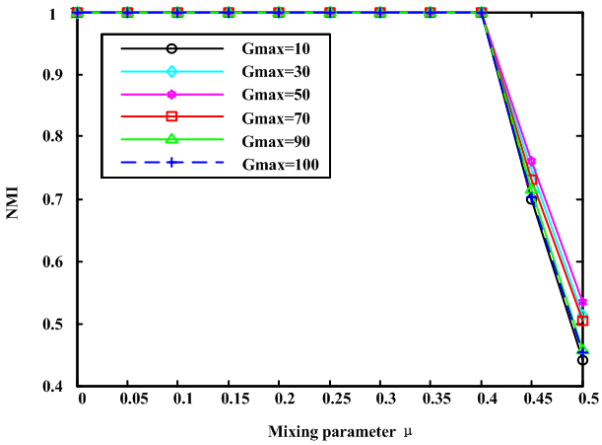


Fig.8 The results of NMI of the extension of GN Benchmark networks over 30 runs with different iteration numbers

Fig.8 shows the max NMI of the extension of GN Benchmark networks over 30 runs varies with different values of G_{max} when choosing initial population size *popsize*=100 and the maximum dominant solution number *NM*=40. Fig.9 and Fig.10 shows the results of NMI and Q and D of six real-world networks respectively.

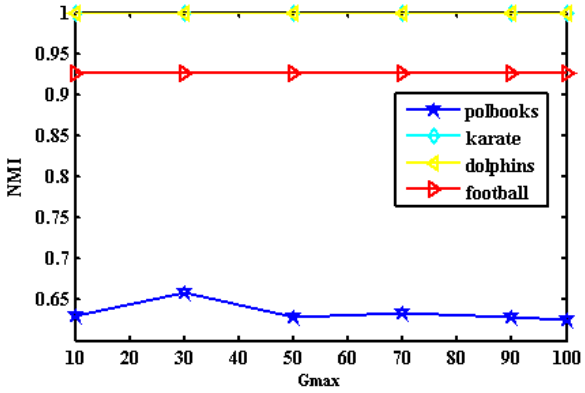


Fig.9 The results of NMI of four real-world networks over 30 runs with different iteration numbers

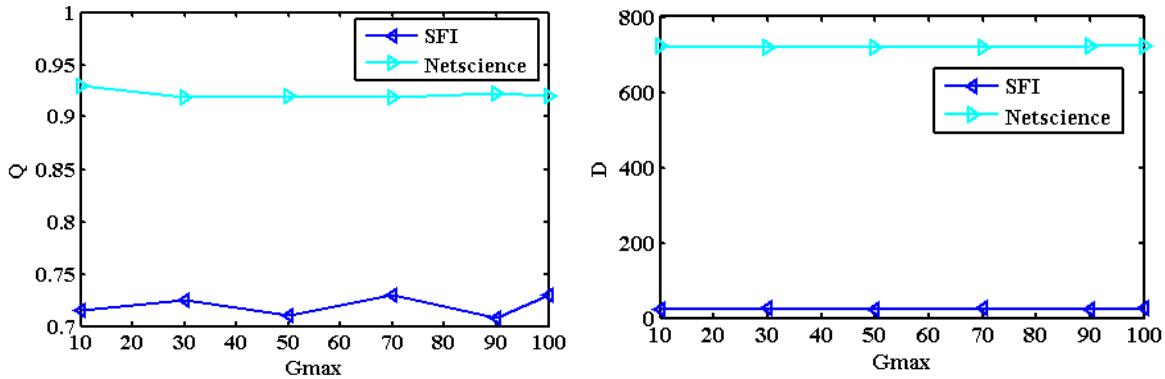


Fig.10 The results of Q and D of two real-world networks over 30 runs with different iteration numbers

As shown in the above three figures, except for polbooks network, the value of G_{max} did not cause significant impact on the result. Therefore the proposed algorithm uses 30 as the maximum number of iterations.

Fig.11 shows how the max NMI of the GN Benchmark networks over 30 runs varies with different values of the maximum dominant solution number *NM* when choosing G_{max}=30 and *popsize* =100. Fig.12 and Fig.13 shows the results of NMI and Q and D of six real-world networks respectively.

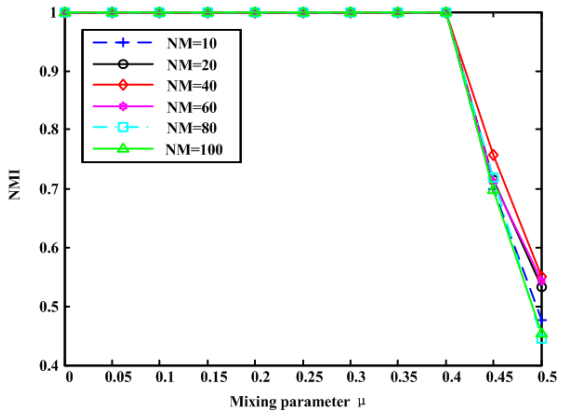


Fig.11 The results of NMI of the extension of GN Benchmark networks over 30 runs with different NM numbers

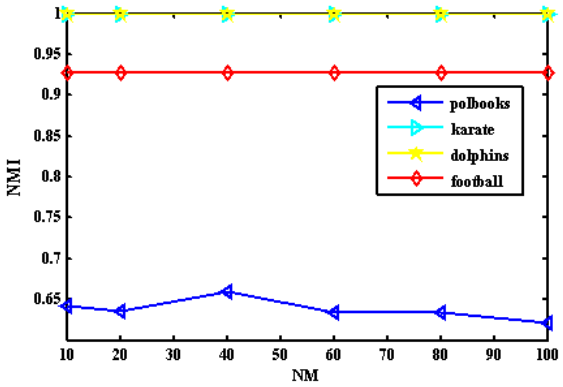


Fig.12 The results of NMI of four real-world networks over 30 runs with different NM numbers

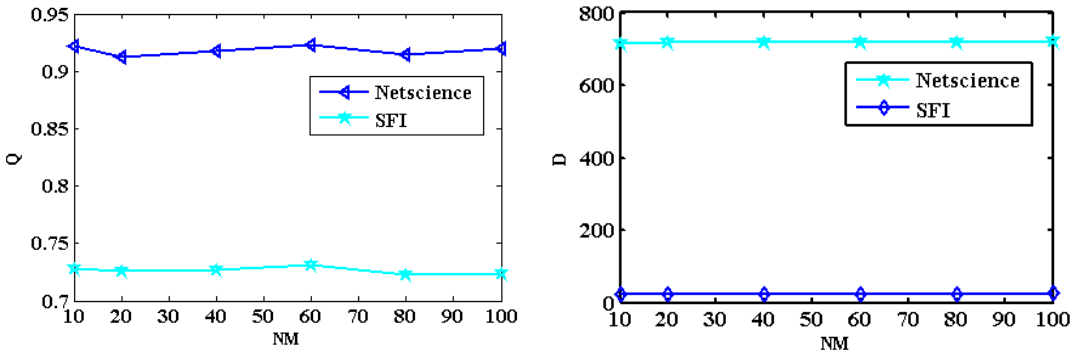


Fig.13 The results of Q and D of two real-world networks over 30 runs with different NM numbers

It can be seen from Fig.11, Fig.12 and Fig.13, that the number of the maximum dominant solution will affect the final algorithm performance. If the value of the maximum dominant solution number *NM* is too small, it will lead to the decline of population diversity and can easily make the algorithm fall into local optima. Choosing a large number of dominant solutions will increase the computational time, and cannot guarantee the proposed algorithm converges to the optimal solution rapidly and efficiently. Thus, from the experimental results shown in the above three figures, the value of NM as 40 is the most suitable.

4. 4.Experimental results and analysis

Here seven algorithms will be tested on the computer-generated networks and eight real-world networks. In the experiments, the maximum generation is 30 and other parameters are the same as in their original papers. For

each network, these algorithms run independently 30 times, and the final results are analyzed. Parameter settings of APMOEA algorithm are shown in TABLE 6:

TABLE 6 Parameters Setting

Parameters	Population size of parameter P N_{umP}	Maximum size of dominant population N_{max}	Maximum number of iterations G_{max}	Crossover probability p_c	Mutation probability p_m
Values	100	40	30	1	0.8

4.4.1. Simulation results on computer-generated networks

In this section, we test the proposed algorithm and other 4 algorithms 30 times on 11 networks, in which the value of mixing parameter μ ranges from 0 to 0.5 with an interval of 0.05. Each of the 11 networks contains 1000 nodes and the cluster size ranges from 10 to 50, τ_1=2 and τ_2=1, and the max node degree is 50, the average node degree is 20. For each algorithm, the maximum number of iterations is 30, and the NMI is selected as the evaluation index. The max value and average value of NMI over 30 runs are shown in Fig.14 and Fig.15 respectively.

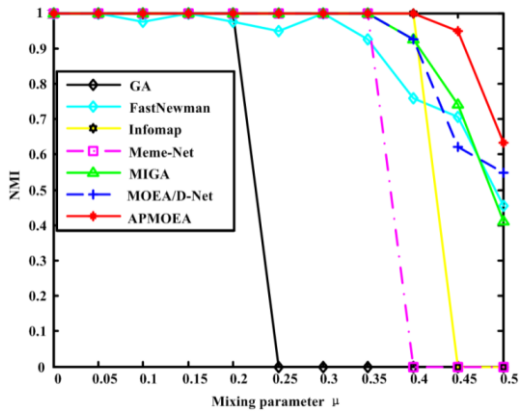


Fig.14 The max value of NMI over 30 runs

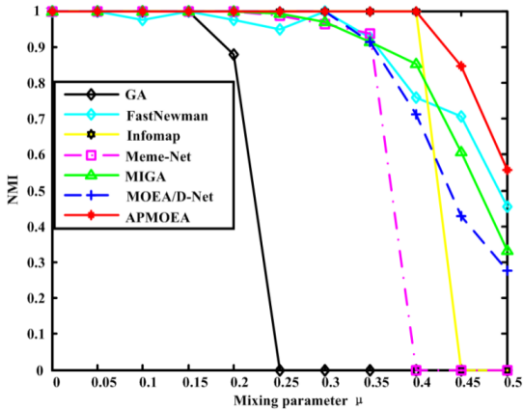


Fig.15 The average value of NMI over 30 runs

Seen from Fig.14 and Fig.15, with the increase of minimum parameter μ, the community structure in networks becomes more and more fuzzy, which makes it more difficult to detect. From Fig.14 we can see that when μ<0.2, all seven algorithms can obtain good results. However, as the value of μ increases to 0.25, the conventional GA algorithm becomes unable to detect the community structure in networks. When μ reaches 0.4, Meme-Net

algorithm begins to produce invalid results, and the values of NMI obtained by FastNewman decrease rapidly after μ reaches 0.3. Infomap works well until μ reaches 0.45. Only the proposed algorithm can find out true partitions after this point. When μ is over 0.45, from Fig.14 we can clearly see that the results detected by APMOEA also has the highest accuracy of all.

As shown in Fig.15, the average value of NMI obtained by the proposed algorithm is still the highest one compared with other algorithms. When μ<0.45, the proposed algorithm can maintain the stability of the detecting results of NMI=1. Even when μ=0.5, the average value of NMI obtained by the proposed algorithm is close to 0.5, while the other results are less than 0.4. From the above analysis, it can be seen that in the detection of the artificial network, the proposed algorithm could find partitions closer to the ground truth with high stability.

As Infomap and FastNewman are non-evolutionary based algorithms, next we will compare the convergence speed of the remaining five evolutionary based algorithms. Fig.16 shows the convergence curves of five algorithms on the artificial network of μ=0.5 in one run.

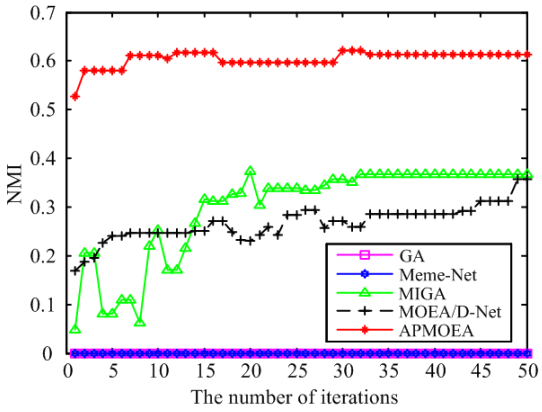


Fig.16 The convergence curves on the artificial network of μ=0.5

As shown in Fig.16, when μ=0.5 the GA algorithm and Meme-Net algorithm are unable to detect the community structure of the network, so in the iterative process their value of NMI is always 0. Here, we will analyze convergence curves of the remaining three algorithms. In about the 30th generation, APMOEA has already converged and obtained an NMI value above 0.6. MIGA algorithm also converges in about the 30th generation, but its final result is less than that of APMOEA. It takes more generations for MOEA/D algorithm to converge and the final result only reaches to about 0.35. It follows that, in the artificial network detection, APMOEA can effectively detect the community structure in complex networks and converges faster.

In order to study the influence of the AP based initialization employed in the proposed algorithm, an experiment is carried out on 11 computer-generated networks using APMOEA, AP algorithm only and a multiobjective evolutionary algorithm (MOEA), which differs from APMOEA by taking random initialization. In the experiment each algorithm is run 30 times respectively, and the best NMI value from 30 results is selected. The experimental results are shown in Fig.17.

Seen from Fig.17, MOEA is unable to achieve ideal results by using random initialization. Because AP algorithm has a high degree of accuracy in dividing networks, thus taking the AP based initialization can greatly

improve the quality of initial population in APMOEA, which boosts the convergence speed of the algorithm. At the same time, compared to the single AP algorithm, APMOEA exploits the advantage of evolutionary algorithms to improve the results and achieves the global optimum. Therefore, compared to AP algorithm and MOEA, the proposed algorithm can get better results.

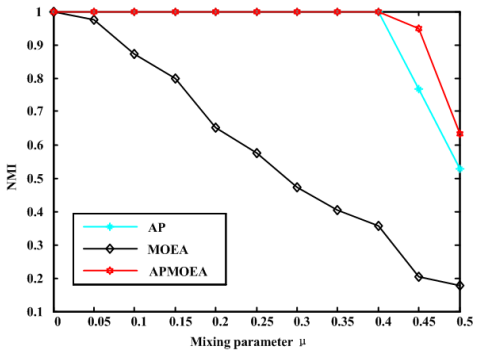


Fig.17 The average value of NMI over 30 runs

4.4.2. Simulation results on real-world networks

In this section, we show the application of APMOEA and six other proposed algorithms on eight real-world networks introduced in 3.1.2. Fig.18 shows a Pareto-optimal front obtained by APMOEA on Zachary's karate club network in one run.

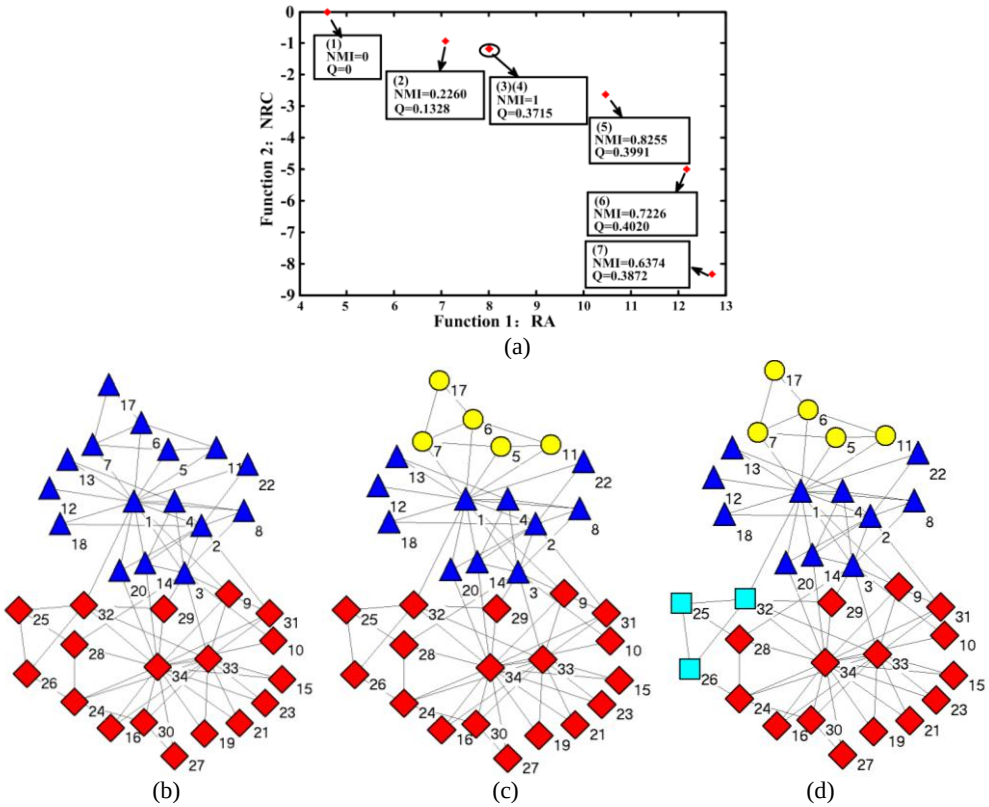


Fig.18 The results on Zachary's karate club network. (a) A Pareto-optimal front of one run. (b) Network corresponding to solution (3) and solution (4). (c) Network corresponding to solution (5). (d) Network corresponding to solution (6)

Fig.18 (a) displays all the solutions in the Pareto-optimal front, and their corresponding values of NMI and *Q* have been indicated in the corresponding boxes. Seen from the networks corresponding to solution (3) to solution

(7), the Pareto-optimal front obtained by APMOEA not only contains the true partition of the network (shown in Fig.18 (b)), but also mines the potential community structure of the network, which achieves the goal of dividing the network into multi hierarchical structures. A big community in Fig.18 (b) is split into two smaller sub-communities, Fig. 18(c), and the new one is marked with ellipse. Fig.18 (d) continues to divide the network into 4 sub-communities, in which more small communities are found. These different kinds of partitions can help us study the network from different points of view and meet the needs of different people. And these partitions can be obtained by the multiobjective optimization algorithm in only one run, which is impossible by using the single objective optimization algorithm.

In the next experiment, each algorithm will run independently 30 times on the first four networks whose true partitions are known, and we employ Normalized Mutual Information (*NMI*), which is introduced in Section 4.2 as the index to estimate the detection results. The maximum number of iterations for the five evolutionary based algorithms is 50. We will record their best results and give an analysis.

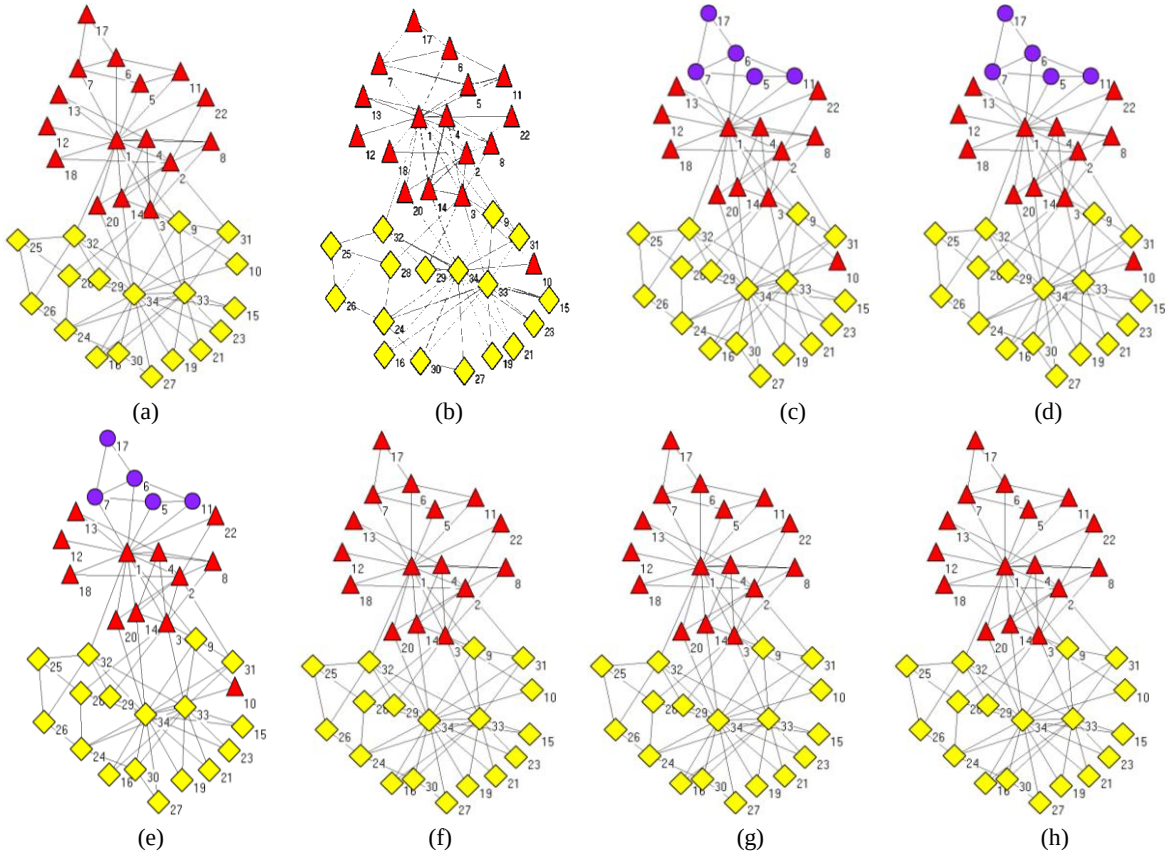


Fig.19 Best detection result out of 30 runs on Zachary's karate club (a) The true partition, (b) (b) Detection result of FastNewman, (c) Detection result of Infomap, (d) Detection result of GA, (e) Detection result of Meme-Net, (f) Detection result of MIGA, (g) Detection result of MOEA/D-Net, (h) Detection result of APMOEA and the detection result of MODPSO.

Fig.19 shows the true partition and the detected results obtained by seven algorithms on Zachary's karate club network. From Fig.19, Infomap, GA and Meme-Net have the same detection results, in which the network is divided into 3 communities and FastNewman partition 10-th node as the opposite group by mistake. While the MOEA/D-Net, MIGA and the proposed algorithm are able to detect the true community structure of the network.

In order to study the convergence of the proposed algorithm, Fig.20 shows the values of NMI obtained by five evolutionary based algorithms (GA, Meme-Net, MIGA, MOEA/D-Net and APMOEA) on Zachary's karate club network in 50 iterations.

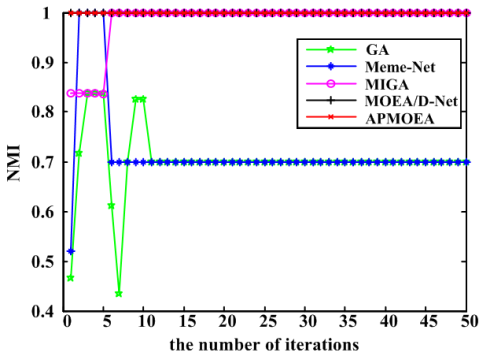


Fig.20 The convergence curves on Zachary's karate club in 30 iterations

From Fig.20 we can see that except for GA and Meme-Net, the values of NMI obtained by the other three algorithms could converge to 1, while the proposed algorithm only requires one iteration for convergence, which is faster than MIGA. Fig.21 shows the true partition and the detected results obtained by seven algorithms on Bottlenose dolphin network.

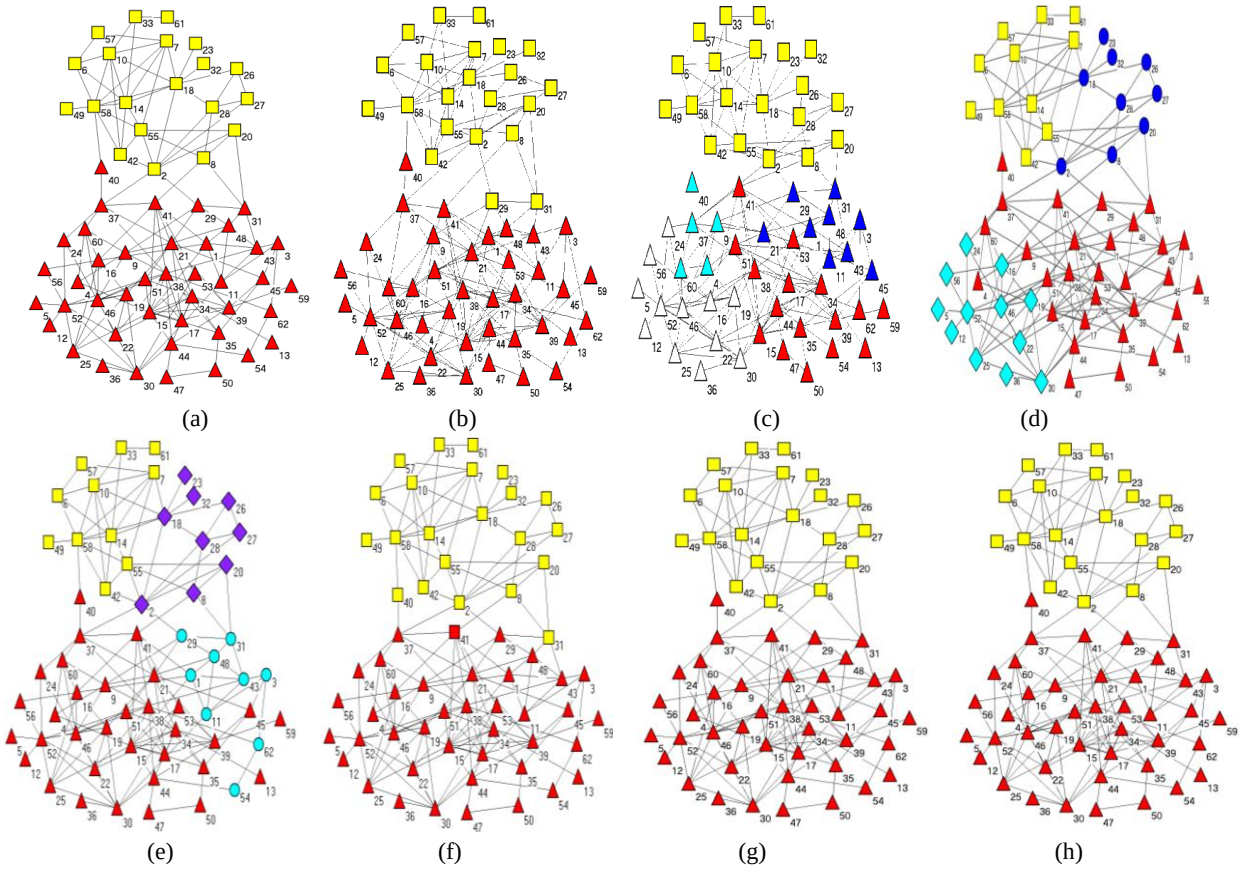


Fig.21 Best detection result out of 30 runs on Bottlenose dolphin network (a) The true partition, (b) Detection result of FastNewman, (c) Detection result of Infomap, (d) Detection result of GA, (e) Detection result of Meme-Net, (f) Detection result of MIGA, (g) Detection result of MOEA/D-Net, (h) Detection result of APMOEA and the detection result of MODPSO.

As shown in Fig.21, Infomap incorrectly divides the network into 5 groups, shown in different colors, and

both GA and Meme-Net divide the network into 4 categories with 21 and 18 nodes being misplaced respectively, while MIGA labels the 31-th node and the 40-th node as the opposite class, MOEA/D-Net and APMOEA detect the true community structure of Bottlenose dolphin network (shown in Fig.21 (g) and Fig.21 (h)), and their correct rate is 100%. Fig.22 shows the convergence curves of five evolutionary based algorithms on Bottlenose dolphin network during 50 iterations. The results show that GA, Meme-Net and MIGA converge to unsatisfactory results before 25 generations, while the proposed algorithm in this paper converges in about 5 generations, and achieves the optimal solution.

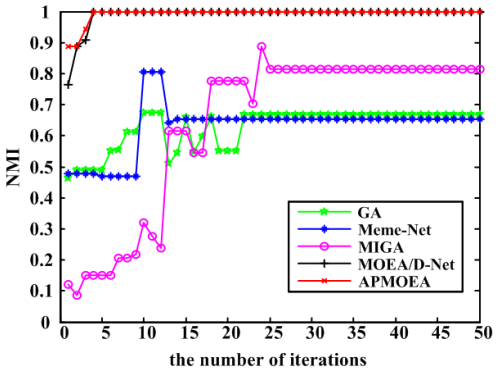


Fig.22 The convergence curves on Bottlenose dolphin in 50 iterations

Fig.23 shows the true partition and the detected results obtained by seven algorithms on American college football network. As can be seen from Fig.23, the detection results obtained by MOEA/D-Net are the best with only 8 points being misplaced. However, it divides the network into 11 categories by mistake. Only the proposed algorithm and MIGA can find the true number of communities.

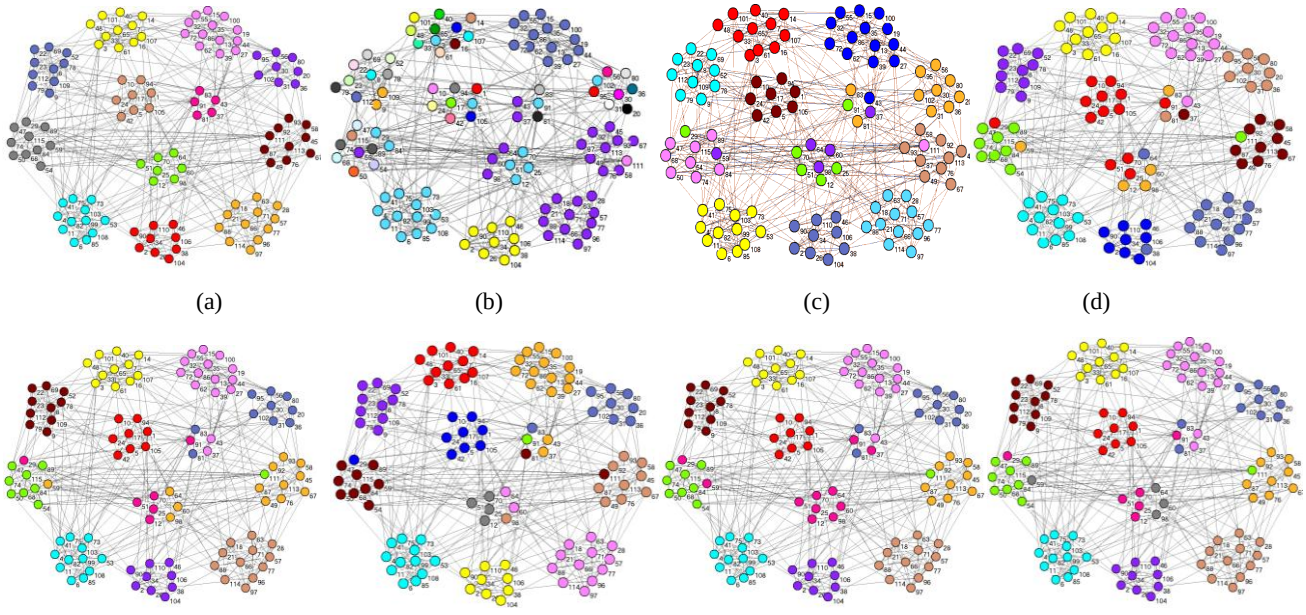


Fig.23 Best detection result out of 30 runs on American college football network (a) The true partition, (b) Detection result of FastNewman, (c) Detection result of Infomap,(d) Detection result of GA, (e) Detection result of Meme-Net, (f) Detection result of MIGA, (g) Detection result of MOEA/D-Net, (h) Detection result of APMOEA and the detection result of MODPSO.

Fig.24 shows the convergence curves of five evolutionary based algorithms on American college football network in 50 iterations. We can see from Fig.24, that GA has the worst performance. MIGA as well as Meme-Net doesn't converge until the 20th generation, and the obtained values of NMI are less than that by APMOEA. Though the final results obtained by MOEA/D-Net exceed that by APMOEA, it takes 45 iterations for MOEA/D-Net to converge. The proposed algorithm has the fastest convergence speed in five algorithms and performs better than GA, MIGA and Meme-Net.

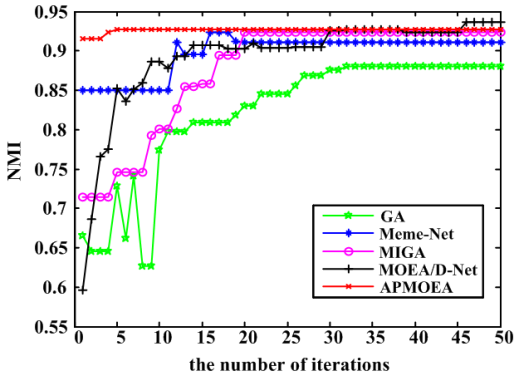
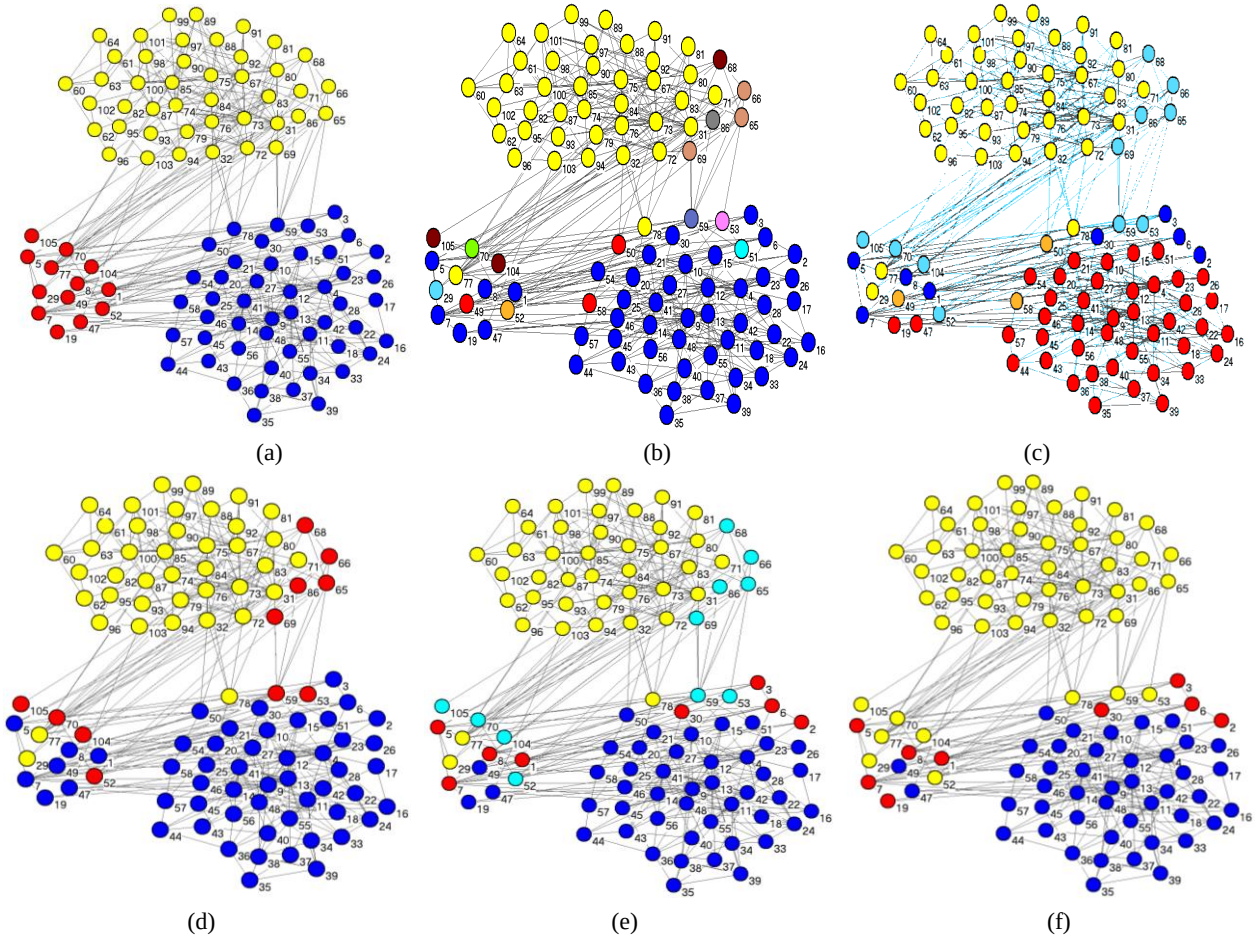


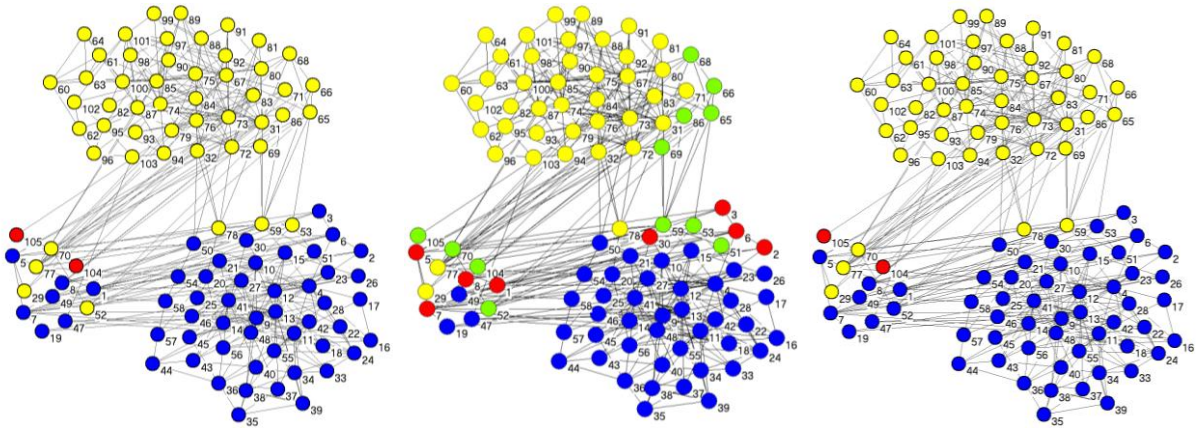
Fig.24 The convergence curves on American college football in 50 iterations

The network of Books about US politics is the hardest one for detecting true partitions. Fig.25 shows the true partition and the detected results obtained by seven algorithms.



(a) (b) (c)

(d) (e) (f)



(g) (h) (i)

Fig.25 Best detection result out of 30 runs on Books about US politics network (a) The true partition, (b) Detection result of FastNewman, (c) Detection result of Infomap, (d) Detection result of GA, (e) Detection result of Meme-Net, (f) Detection result of MIGA, (g) Detection result of MOEA/D-Net, (h) Detection result of MODPSO, (i) Detection result of APMOEA

It can be seen from Fig.25 that all algorithms can find the correct number of communities except for Meme-Net, which divides the network into 3 categories. Nevertheless, the result obtained by APMOEA only misplaced 13 points, which has the least error rate.

Fig.26 shows the convergence curves of five evolutionary based algorithms on Books about US politics network in 50 iterations. From Fig.26 we can see that although GA and Meme-Net converge faster than APMOEA, their detection results are less good. Result obtained by Meme-Net shows significant errors. MIGA performs better than GA and Meme-Net, but its convergence speed is slow, and the final result is not ideal. MOEA/D-Net reaches convergence in about the 32nd generation and its accuracy is 0.621. While APMOEA has the best performance during the whole iterative process and achieves a final result whose NMI value is 0.659 in about the 18th generation. This experimental evidence demonstrates the effectiveness of the proposed algorithm.

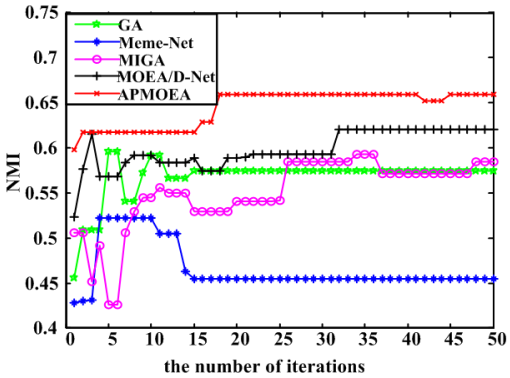


Fig.26 The convergence curves on Books about US politics network in 50 iterations

Next, we will test these algorithms on the remaining four networks with unknown ground truths. Since MIGA algorithm requires the use of a priori information, namely the real number of communities in a network which is unknown in these problems, we only consider the other six algorithms here. At the same time, modularity Q introduced in formula (6) will be used as the metric index instead of NMI, which only works for networks when their true partitions are known.

Here we take netscience network as representative to analyze the convergence of several evolutionary algorithms. Taking Climbing Hill as its local search strategy, which is time-consuming, Meme-Net algorithm is unable to output the final results after many iterations on netscience network, therefore Fig.27 shows the convergence curves of the remaining three algorithms on that network in 51 iterations.

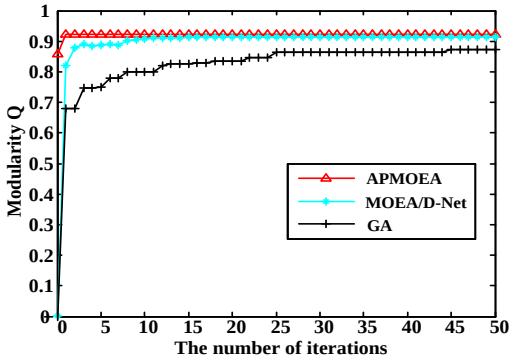


Fig.27 The convergence curves on netscience network in 50 iterations

As we can see from Fig.27 the 0-th iteration means the accuracy of the partition results of the pre-processing procedures of those algorithms. GA employs only stochastic sequence as the initialization, which leads to meaningless partition results and its value of modularity Q is close to 0. Though MOEA/D-Net takes the initialization strategy based on the neighboring information of each node, it also has randomness and plays little role in medium-scale network. The proposed algorithm employs AP as the pre-processing step, which is much more accurate and obtains higher modularity Q. On the basis of these effective pre-partition results, APMOEA converges to a good value early on, while it takes nearly 15 generations for MOEA/D-Net to achieve stability and GA doesn't converge until 25 generations. This is possibly because the proposed algorithm uses the AP based initialization method to obtain better initial population, which leads to a faster convergence speed compared with other evolutionary algorithms.

Fig.28 shows the results obtained by GA, MOEA/D-Net and APMOEA on Power grid network from iteration 0 to iteration 50. It can be seen from the curves that the initializations of the three algorithms matter little and the results of modularity Q are all below 0.1, but AP gets better results and it helps APMOEA algorithm converge to the best result quickly, while MOEA/D-Net achieves stability in about 10 iterations with a moderate result. GA does not converge until close to 50 generations with a result of 0.65.

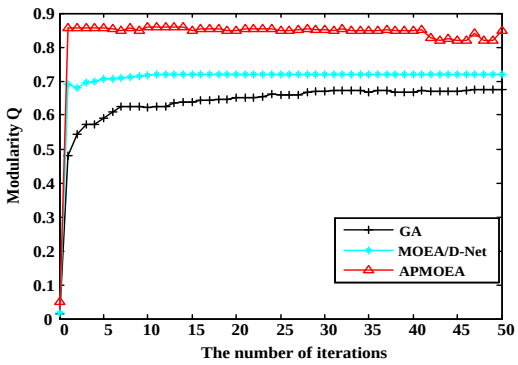


Fig.28 The convergence curves on Power grid network in 50 iterations

TABLE 7 shows the best detection results of NMI obtained by FastNewman(Alg1), Infomap(Alg2), GA(Alg3), Meme-Net(Alg4), MIGA(Alg5), MOEA/D-Net(Alg6), MODPSO(Alg7) and APMOEA(Alg8) algorithms on the first four real-world networks whose true partitions are known in 30 runs and TABLE 8 shows the best detection results of *Q* obtained by the six algorithms mentioned above except for MIGA, which needs prior information, on the remaining four networks whose true partitions are unknown in 30 runs. Here symbol "×" means the algorithm has no detection results and symbol "——" means the algorithm cannot give its result after many iterations.

TABLE 7 The best values of NMI obtained by eight algorithms in 30 runs

Network	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7	Alg8
Zachary's Karate Club	0.837	0.699	0.699	0.699	**1**	**1**	**1**	**1**
Bottlenose Dolphins	0.814	0.587	0.667	0.687	0.814	**1**	**1**	**1**
American College football	0.710	0.924	0.881	0.911	0.916	**0.937**	0.927	0.927
Books about US politics	0.588	0.537	0.575	0.554	0.585	0.621	0.598	**0.659**

It can be seen from TABLE 7 that APMOEA can find the true partition on Zachary's Karate Club network and Bottlenose Dolphins network. Although the result on American College football obtained by APMOEA is slightly worse than that by MOEA/D-Net, it is better than the others. Furthermore, the solution obtained in Books about US politics network is much better than those of the other four proposed algorithms. This suggests that the proposed algorithm has better performance, especially on some networks with fuzzy community structure.

TABLE 8 The best values of *Q* obtained by seven algorithms in 30 runs

Network	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7	Alg8
SFI	0.734	0.733	0.587	0.710	×	0.731	**0.748**	0.739
netscience	——	0.931	0.858	——	×	0.914	**0.950**	0.923
Power grid	——	0.830	0.666	——	×	0.688	0.842	**0.858**
PGP	——	**0.813**	0.645	——	×	0.676	0.335	0.726
Internet	——	**0.576**	0.454	——	×	——	——	0.516

TABLE 8 shows that the proposed algorithm achieves the best values of modularity *Q* in power grid networks, but fails to exceed Infomap in Internet network and PGP network. However, APMOEA can still obtain better results compared with most of the other algorithms, especially as some of them are incapable of calculating their results after many iterations. Generally speaking, the proposed algorithm could detect better results in the eight real-world networks. These results strongly suggest the effectiveness of the proposed algorithm.

TABLE 9 shows the detection results of *D* obtained by eight algorithms on seven networks. Symbol "×" means the algorithm has no detection results and "——" means the algorithm cannot give its result in a long time.

TABLE 9 The best values of *D* obtained by eight algorithms in 30 runs

Network	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7	Alg8
Karate	6.022	7.845	6.833	**7.845**	6.823	**7.845**	7.842	**7.845**
Dolphins	7.817	10.075	9.095	11.707	11.209	11.725	10.810	**11.784**
Football	28.406	42.846	13.538	44.340	24.455	**44.388**	40.165	43.348
SFI	20.727	25.632	19.697	24.932	×	25.225	**26.043**	25.753
Netscience	622.877	718.716	452.755	——	×	700.934	727.82	**720.22**
Power grid	104.086	691.936	121.07	——	×	692.493	664.231	**710.143**
PGP	——	1021.14	124.64	——	×	——	1032.93	**1040.2**

It can be seen from TABLE 9 that on the most of networks, the proposed algorithm can achieve relatively high value of *D*.

TABLE 10 shows the number of communities detected by eight algorithms on first four real-world networks.

TABLE 10 The number of communities detected by eight algorithms on four real-world networks

Network	Alg1	Alg2	Alg3	Alg4	Alg5	Alg6	Alg7	Alg8
Zachary's Karate Club	**2**	3	3	3	**2**	**2**	**2**	**2**
Bottlenose Dolphins	**2**	5	3	4	**2**	**2**	**2**	**2**
American College football	49	**12**	11	11	**12**	11	**12**	**12**
Books about US politics	12	5	**3**	4	**3**	**3**	4	**3**

It can be seen from TABLE 10, only the proposed algorithm and MIGA can find the correct numbers of communities of all the networks. In addition, the overall partitioning results obtained by the proposed algorithm are better than that of MIGA. Combined with the analysis in TABLE 7, it can be seen that the proposed algorithm can find the partition which is closest to the true community structure of networks.

4.4.3. Simulation results on real-world networks of APMOEA and APEA

In this section, we will show the application of APMOEA and a single-objective evolutionary algorithm (we take modularity density D as the objective) based on the preliminary partitions attained by AP algorithm (which is hereinafter referred to as APEA) on six real-world networks introduced in 3.1.2. Fig.29 shows the best values of NMI for APMOEA and APEA attained on four real-world networks for which the true partitions are known.

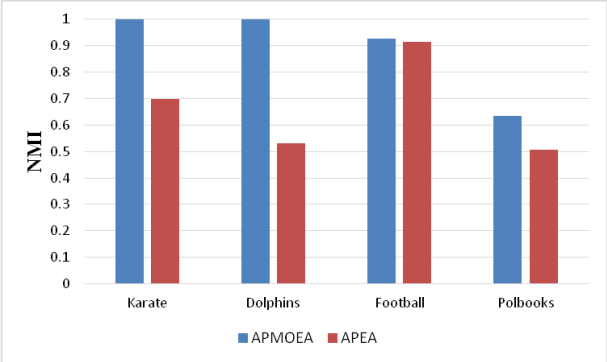
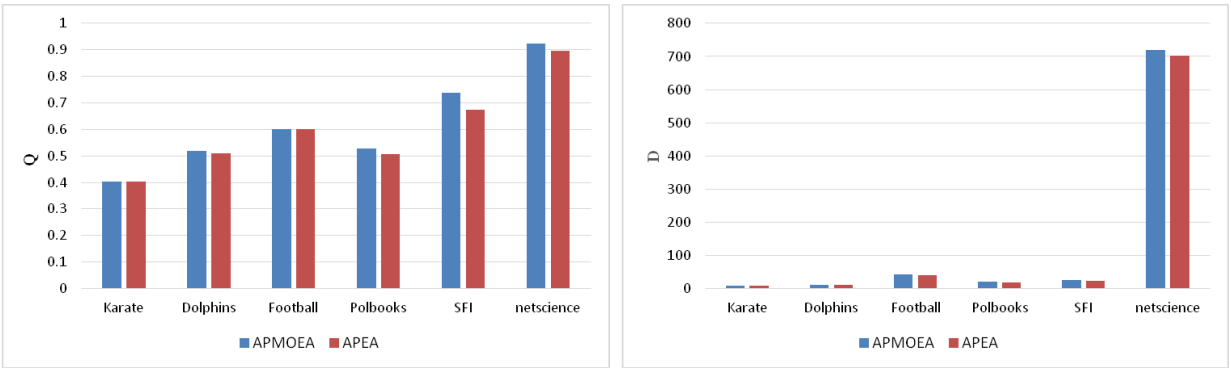


Fig.29 the best values of NMI of APMOEA and APEA attained on four real-world networks

As shown in Fig.29, APMOEA can acquire more accurate partitions of networks than APEA. The multiobjective evolutionary algorithm shows advantages over the single-objective evolutionary algorithms when they are all based on the preliminary partitions attained by AP algorithm.

Fig.30 (a) and Fig.30 (b) shows the best results of Q and D of APMOEA and APEA on six real-world networks. Fig.30 indicates that: our proposed algorithm can acquire higher Q and D on most of real-world networks than APEA; multiobjective optimization can achieve a group of nondominated solutions in one run and it reveals the hierarchical structure of networks to meet different needs for division; the optimal solutions found by single objective optimization are usually included in the Pareto-optimal set. This provides further evidence that the multiobjective evolutionary algorithm has advantages over single-objective evolutionary algorithms when they are all based on the same preliminary partitions attained by AP algorithm.



(a) (b)

Fig. 30 the best results of Q and D of APMOEA and APEA on six real-world networks. (a) the best results of Q of APMOEA and APEA on six real-world networks. (b) the best results of D of APMOEA and APEA on six real-world networks.

4.4.4. The average total time of the eight algorithms

TABLE 11 shows the average total time of eight algorithms, the data in bold represent that the algorithm costs more time than the proposed algorithm on the corresponding networks.

TABLE 11 The average total time of the eight algorithms

Network	Alg1(s)	Alg2(s)	Alg3(s)	Alg4(s)	Alg5(s)	Alg6(s)	Alg7(s)	Alg8(s)
Karate	0.196	0.845	2.956	2.081	4.671	**516.469**	12.875	17.797
Dolphins	0.817	1.474	8.336	13.163	7.114	**879.92**	21.218	31.212
Football	4.956	1.935	34.904	**113.569**	11.496	**1062.9**	47.015	50.950
SFI	5.242	3.864	**154.043**	**369.841**	×	**1529.01**	38.922	97.065
Netscience	**22154.5**	724.036	1036.6	——	×	**113382.1**	1282.16	14008.98
Power grid	**1379828.8**	8211.508	4810.02	——	×	**1353557.5**	8830.64	95390.08
PGP	——	24073.3	25269.22	——	×	——	34792.9	426212.9

As shown in TABLE 11, the proposed algorithm costs more time than Alg1~Alg5, and Alg7 on the first two networks, but along with the increase of the size of the network data, the proposed algorithm costs less time than Alg1, Alg4, Alg5, and Alg6. For the average total time, among 8 algorithms, the proposed algorithm is not the best and not the worst.

5. Conclusion

This paper has presented a multiobjective evolutionary algorithm based on affinity propagation to solve community detection problems. Firstly, the algorithm employs a similarity measure based on signal transmission to transform the graph clustering problem into a data clustering problem, and uses the AP method to obtain a set of preliminary partitions of the network. As AP method has high accuracy and fast clustering speed, we can make use of it to get satisfying satisfactory preliminary partition results within a few steps. Next, those AP solutions are taken as the initial population of the multiobjective evolutionary algorithm, in which the set of Pareto-optimal solutions will be updated through constantly selecting the nondominated ones from the population after crossover and mutation. Through the above steps, the diversity of the population is increased, thereby improving the likelihood of

getting better overall partition results. Finally, these two parts of solutions will be merged into one, from which the final Pareto-optimal solutions are chosen. The proposed method not only takes advantage of the AP method to quickly find a set of superior initial solutions, but also uses the characteristic of multipoint searching in multiobjective evolutionary algorithm for a further search to reach the global optimum. Through the effective combination of these two components, AP clustering methods and multiobjective evolutionary algorithm, we can quickly pre-treat the network through data clustering method and then use the evolutionary algorithm to search for globally optimal solutions. Experimental results have shown that in most of the networks, APMOEA has faster convergence rate as well as more accurate detection results compared with other algorithms. Meanwhile, the proposed algorithm can also mine the multi-hierarchical structure of networks.

Nevertheless, there is still room to improve the proposed algorithm in future work. The experiments shown in TABLE 7, suggest that APMOEA sometimes does not work very well in some networks with medium-large size, such as PGP, and the selection process of parameter *P* or the choice of nondominated solutions is somewhat time consuming. Future work will consider how to simplify the process of generating the preliminary partitions by using AP method, to make the proposed algorithm more suitable for partitioning networks and modify the local search strategy or employ some helpful ways to make the proposed algorithm more accurate in the detection of large networks.

Acknowledgement

We would like to express our sincere appreciation to the anonymous reviewers for their valuable comments, which have greatly helped us in improving the quality of the paper. This work was partially supported by the National Basic Research Program (973 Program) of China under Grant 2013CB329402, the National Natural Science Foundation of China, under Grants 61371201 and 61272279, the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170, and the EU FP7 project (grant no. 247619) on "NICaiA: Nature Inspired Computation and its Applications".

References

[1] D. J. Watts, S. H. Strogatz, Collective dynamics of "small-world" networks. Nature 393(6684) (1998) 440-442.

[2] A. L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286(5439) (1999) 509-512.

[3] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2004) 026113.

[4] D.W. Zhang, F.D. Xie, Y. Zhang, F.Y. Dong, K. Hirota, Fuzzy analysis of community detection in complex networks, Physica A, 389, (2010): 5319-5327.

[5] J.S. Wu, Y.T. Hou, Y. Jiao, Y. Li, X.X Li, L.C. Jiao, Density shrinking algorithm for community detection with path based similarity, Physica A, 433 (2015): 218-228.

[6] S. Fortunato, V. Latora, M. Marchiori, A method to find community structures based on information centrality,

Phys. Rev. E 70 (2004) 056104.

[7] B. M. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal 49 (2) (1970) 2912307.

[8] F. Wu, B. A. Huberman, Finding communities in linear time: a physics approach, Eur. Phys. J. B 38 (2004) 331-338.

[9] F.D. Xie, M. Ji, Y. Zhang, D. Huang, The detection of community structure in network via an improved spectral method, Physica A 388 (2009) 3268-3272.

[10] Y. Pan, D. H. Li, J.G. Liu, J.Z. Liang, Detecting community structure in complex networks via node similarity, Physica A 389 (2010) 2849-2857.

[11] H.W. Shen, X.Q. Chen, C. Cai, M.B. Hu, Detect overlapping and hierarchical community structure in networks, Physica A 388 (2009) 1706-1712.

[12] Z.H. Wu, Y. F. Lin, H.Y. Wan, S.F. Tian, K.Y. Hu. Efficient overlapping community detection in huge real-world networks, method, Physica A 391(2012) 2475-2490.

[13] J.S. Wu, X.H. Wang, L.C. Jiao, Synchronization on overlapping community network, Physica A 391 (2012) 508-514.

[14] R. Guimerà, M.Sales-Pardo, L. A. N. Amaral, Modularity from fluctuations in random graphs and complex networks, Phys. Rev. E 70 (2) (2004) 025101 (R).

[15] M. E. J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (2006) 8577-8582.

[16] M. E. J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.

[17] A. Clauset, M. E. J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (2004) 066111.

[18] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Phys. Rev. E 72 (2005) 027104.

[19] J.Q. Jiang, L.J. McQuay, Modularity functions maximization with nonnegative relaxation facilitates community detection in networks, Physica A 391 (2012) 854-865.

[20] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (2010) 75-174.

[21] C. Pizzuti, Ga-net: a genetic algorithm for community detection in social networks, in: Parallel Problem Solving from Nature C PPSN X, in: Lect. Note Comput. Sc., vol. 5199, Springer, Berlin, Heidelberg, 2008, pp. 1081-1090.

[22] C. Pizzuti, A multi-objective genetic algorithm for community detection in networks, in: Proceedings of the 21st IEEE International Conference on Tools with Artificial Intelligence, Newark, New Jersey, USA, 2009, pp. 379-386.

[23] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182-197.

[24] M. G. Gong, B. Fu, L. C. Jiao, and H. F. Du, Memetic algorithm for community detection in networks, Phys. Rev. E 00 (2011) 006100.

[25] Z. Li, S. Zhang, R. S. Wang, X. S. Zhang, L. Chen, Quantitative function for community detection, Phys. Rev. E 77 (2008) 036109.

[26] R. H. Shang, J. Bai, L. Jiao, C. Jin, Community detection based on modularity and an improved genetic algorithm, Physica A 392 (2013) 1215-1231.

[27] M. G. Gong, L. J. Ma, Q. F. Zhang, L. C. Jiao, Community detection in networks by using multiobjective evolutionary algorithm with decomposition, Physica A 391 (15) (2012) 4050-4060.

[28] M. Gong, Q. Cai, X. Chen, L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, 2013, http://dx.doi.org/10.1109/TEVC.2013.2260862.

[29] Q. Zhang, H. Li, Moea/d: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712-731.

[30] Q. Huang, T. White, G.B. Jia, M. Musolesi, N. Turan, K. Tang, S. He, J. K. Heath, X. Yao, Community Detection Using Cooperative Co-evolutionary Differential Evolution, In Proceedings of the 12th International Conference on Parallel Problem Solving from Nature (PPSN XII). Springer. Taormina, Italy. Lecture Notes in Computer Science 7492. September 2012.

[31] B. J. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (2007) 972-976.

[32] Y. Xiao, J. Yu, Semi-supervised clustering based on affinity propagation [J], Journal of Software 19 (11) (2008) 2803-2813.

[33] K. J. Wang, J. Zhang, D. Li at el, Adaptive affinity propagation clustering [J], Acta Automatica Sinica 33(12) (2007) 1242-1246.

[34] M. Gong, L. Jiao, H. Du, L. Bo, Multiobjective immune algorithm with nondominated neighbor-based selection, Evolutionary Computation, MIT Press 16 (2) (2008) 225-255.

[35] J. Handle, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE transactions on Evolutionary Computation 1(1) (2007) 56-76.

[36] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. USA 104 (2007) 36-41.

[37] L. Angelini, S. Boccaletti, D. Marinazzo, M. Pellicoro, S. Stramaglia, Identification of network modules by optimization of ratio association, Chaos 17 (2) (2007)023114.

[38] Y.-C. Wei, C.-K. Cheng, Ratio cut partitioning for hierarchical designs, IEEE Trans. Comput. Aid. D 10 (7) (1991) 911-921.

[39] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, University of California Press, Berkeley, 1967, pp. 281.

[40] F, Ding, Z. Luo, J. Shi, X. Fang, Overlapping community detection by kernel-based fuzzy affinity propagation, Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on.

[41] C. Jia, Y. Jiang, J. Yu, Affinity propagation on identifying communities in social and biological networks, in: Proceedings of The Fourth International Conference on Knowledge Science, Engineering and Management, KSEM'2010, Sep. 1-3, Belfast, UK, LNAI Springer, Heidelberg, 2010.

[42] D. Lai, C. Nardini, H. Lu, Partitioning networks into communities by message passing, Phys. Rev. E 83 (2011) 016115.

[43] S. Yang, Community detection based on adaptive kernel affinity propagation, Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on.

[44] S. E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (2007) 27-64.

[45] Y.-W. Jiang, C.-Y. Jia, J. Yu, Community detection in complex networks based on vertex similarities, Computer Science 38 (7) (2011).

[46] Y.-Q. Hu, M.-H. Li et al, Community detection by signaling on complex networks, Phys. Rev. E 78 (2008) 016115.

[47] K. C. Tan, Y. J. Yang, C. K. Goh, A distributed cooperative coevolutionary algorithm for multiobjective optimization, IEEE Trans. on Evolutionary Computation, 10 (5) (2006) 527-549.

[48] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.

[49] W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33(1977) 452-473.

[50] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of Doubtful Sound features a large Proportion of long-lasting associations, Behavioral Ecologyand Sociobiology 54 (2003) 396-405.

[51] L.Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Metch 78 (2005) P09008.

[52] M. Boguñá, R. Pastor-Satorras, A. Díaz-Guilera, A. Arenas, Models of social networks based on social distance attachment. Phys. Rev. E **70** (2004) 056122.

[53] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (2006) 036104.

[54] M. Rosvall, C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, Proc. Natl. Acad. Sci. USA, (2008) 1118-1123.

Highlights

1. An algorithm called APMOEA is presented for community detection.
2. APMOEA takes affinity propagation to preliminarily divide the network.
3. APMOEA selects nondominated solutions as its initial population.
4. APMOEA finds solutions approximating to the true Pareto optimal front.
5. APMOEA uses an elitist strategy to prevent the degeneration.